

**WEIGHTED ERROR/ERASURE CORRECTION IN A MULTI-TRACK STORAGE
MEDIUM**

1. Field of the Invention:

5 The present invention relates generally to the
application of error correcting codes to storage media.
Specifically, the present invention provides a method,
computer program product, and data processing system for
correcting errors and erasures in a storage medium having
10 multiple tracks of data.

2. Background of the Invention:

15 Error correcting codes (ECCs) play a vital role in
today's world. As storage systems pack more and more
information into smaller physical spaces, as wireless and
other communications systems transmit data over cluttered
communications media, as our technology pushes the limits
of physics, there is an ever-increasing need to be
protected from data errors. Compact discs, magnetic
20 tapes and disks, memory chips, and satellite
transmissions are among the many technologies that are
prone to error from physical forces, yet we take these
things for granted. The Voyager spacecraft developed by
NASA in the 1970s, for instance, transmits radio signals
25 to Earth with less power than emitted by an ordinary 60-
watt light bulb, yet it has captured some of the clearest
pictures of the outer planets taken to date.

30 A remarkable technology makes all of this
reliability possible. All of these technologies rely, at
least in part, on the use of error-correcting codes.

Error correcting codes add a sufficient level of redundancy to a piece of data to be able to recover the data should the data become corrupted. Many such error correcting codes have been developed, such as Hamming codes, which are described in Hamming, R.W., Error detecting and error correcting codes, *Bell System Tech. J.*, 26 (1950), pp. 147-160.

Many modern applications that make use of error correcting codes use what are known as Reed-Solomon codes, including the Voyager spacecraft and compact discs (which use a variant of Reed-Solomon codes known as CIRC, Cross-Interleaved Reed-Solomon Codes). Reed-Solomon codes were first described in Reed, I.S. and G. Solomon, Polynomial Codes over Certain Finite Fields, *J. Soc. Ind. Appl. Math.*, 8 (1960), pp. 300-304.

The Reed-Solomon codes are part of a family of codes known as Bose-Chaudhuri-Hocquenghem (BCH) codes. BCH codes have the desirable property that they can correct a large number of errors with a minimum of redundant information; BCH codes can also correct a larger number of errors if the locations of the errors are known in advance. Errors with known locations are called "erasures" in the art. Also, in the art, an "error" is an error with an unknown location. Several algorithms, such as the Berlekamp-Massey algorithm, are known to those skilled in the art for efficiently decoding BCH codes to recover erasures and errors. Several of these algorithms are described in Blahut, Richard E., The Theory and Practice of Error Correcting Codes, Addison-Wesley, Reading, MA (1983), pp. 161-206.

Mathematically, a Reed-Solomon code maps values from a vector space of a first dimension over a finite field to a vector space of a second, higher dimension over the same field. The values in the second vector space
5 correspond to coefficients of a set of linear equations, the solution to which is the data to be recovered. The element of redundancy in a Reed-Solomon code stems from the fact that the number of linear equations provides exceeds the minimum number "m" needed to recover the
10 data, and that any "m" of the equations are linearly independent. In other words, you can recover the data by solving any "m" of the equations as a system.

Algorithms such as the Berlekamp-Massey algorithm described previously, perform this decoding step
15 efficiently, taking into account the known locations of erasures (that is, which of the linear equation coefficients have been corrupted) so as to be able to recover a greater amount of data when erasures can be identified.

20 In any Reed-Solomon code, the entire vector of data can be recovered only if

$$2t + e \leq n - m$$

where "t" is the number of errors, "e" is the number of erasures, "n" is the number of linear equations available
25 to choose from (i.e., the dimension of the second vector space), and "m" is the minimum number of linear equations necessary to recover the data. In other words, it takes one more uncorrupted linear equation to find an error than it does to simply correct one.

Docket No. 2001-010-TAP

Reed-Solomon codes, because they rely on the use of multiple sets of coefficients, are easily adapted for use in multi-track recording media, such as magnetic tape. An encoded vector of data can be made to span multiple
5 tracks, such that the coefficients for each of the Reed-Solomon equations reside on a separate track. In that way, if one or more tracks become corrupted, but at least "m" tracks can be read successfully, the entire original vector of data can be recovered. Of course, identifying
10 those "m" tracks is easier when some of the corrupted tracks are already known.

Error detecting codes provide a way of identifying errors in a stream of data. One very effective error-detecting code is the cyclic redundancy check (CRC). CRC
15 codes are described in Messmer, Hans-Peter, The Indispensable PC Hardware Book, 2d. Ed., Addison-Wesley, Reading, MA (1995), pp. 694-702. An erasure on a particular track can be identified by interposing CRC codes on each track at periodic intervals. The CRC codes
20 act as a sort of checksum for the data they follow. An erasure can be identified by comparing a CRC code calculated from a block of data on a given track with the CRC code recorded at the end of the block.

While this is an accurate way to identify a track
25 containing corrupted data, it says nothing about the location within the block of data or the extent of the corruption. Thus, a section of magnetic tape with small amounts of corruption on all or most of the tracks will appear as all containing erasures simultaneously, when in
30 fact, there may only be isolated incidents of corruption

Docket No. 2001-010-TAP

on different tracks at different times. If, in such a situation, all of the tracks known to contain errors are treated as erasures, no data will be recovered, since the erasure tracks will be disregarded as corrupted, and
5 there will not be enough remaining tracks to be able to recover the data.

Thus, while identification of erasures is helpful when large amounts of data on a given track are corrupted, it can cause problems when small errors are
10 randomly distributed across many of the tracks. It would, thus, be desirable to have a system for correcting errors in a multi-track medium that is adapted to handle both large erasures and small errors.

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2

SUMMARY OF THE INVENTION

The present invention provides a method, computer
5 program product, and data processing system for
recovering corrupted data in a multi-track storage
medium, such as magnetic tape, using a Reed-Solomon or
other similar error correcting code system. A correction
history keeping a tally of actual corrected instances of
10 corruption in the storage medium is established. The "N"
tracks requiring the greatest number of actual error
corrections are designated as erasures. Any errors in
the remaining tracks must be found on an individual basis
before being corrected.

U.S. PAT. & TM. OFFICE

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of
5 the invention are set forth in the appended claims. The
invention itself, however, as well as a preferred mode of
use, further objectives and advantages thereof, will best
be understood by reference to the following detailed
description of an illustrative embodiment when read in
10 conjunction with the accompanying drawings, wherein:

Figure 1 is an external view of a data processing
system in which the processes of the present invention may
be performed;

Figure 2 is a block diagram of such a data processing
15 system;

Figure 3 is a diagram depicting multi-track magnetic
tape;

Figure 4 is a diagram depicting a process of error
correction in accordance with a preferred embodiment of
20 the present invention;

Figure 5 is a diagram illustrating how the processes
of a preferred embodiment of the present invention may be
applied to a multi-track data storage medium to maximize
error-correcting capabilities;

Figure 6 is a diagram depicting a process of
25 decoding one block of Reed-Solomon encoded data from a
multi-track magnetic tape in accordance with a preferred
embodiment of the present invention;

Figure 7 is a flowchart representation of a process of correcting errors in a multi-track storage medium in accordance with a preferred embodiment of the present invention;

5 **Figure 8** is a flowchart representation of a process of error correction code decoding a block of data for a multi-track medium using a variable number of erasures in accordance with a preferred embodiment of the present invention; and

10 **Figure 9** is a block diagram of special-purpose
hardware for performing error correction in an
alternative embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Figure 1 is a diagram representing an overall view
5 of a multi-track recording device (tape drive 100) in
accordance with a preferred embodiment of the present
invention. The present invention provides a method,
computer program product, and data processing system for
correcting errors in a multi-track data storage medium.
10 One commonly used multi-track data storage medium is
magnetic tape. The embodiment herein disclosed utilizes
magnetic tape as the physical data storage medium, but
one of ordinary skill in the art will recognize that the
present invention is applicable in conjunction with any
15 physical data storage medium utilizing multiple tracks of
data.

Tape drive 100 reads and writes data to/from
magnetic tape 102. Data buffer 104 acts as a temporary
storage location for data read from or being written to
20 magnetic tape 102. Write circuitry 106 reads data from
data buffer 104, processes the data, and transmits the
processed data to read/write head 108 for writing to
magnetic tape 102. Write circuitry 106's processing of
the data includes encoding the data with an error
25 correcting code, such as a Reed-Solomon code, and may
include additional processes such as appending error-
detecting codes such as CRC codes to the data.

Read circuitry 110 performs what is essentially the
opposite transaction as write circuitry 106. Read
30 circuitry 110 decodes data read from magnetic tape 102 by

Docket No. 2001-010-TAP

read/write head 108, correcting errors as necessary, and writes the data to data buffer 104.

In a preferred embodiment of the present invention, tape drive 100 is associated with a host computer or
5 network (not shown), and data buffer 104 is read from and written to by the host computer or by other devices (such as computers or network-connected peripherals). For example, a host computer may write data to data buffer 104, which is then written by tape drive 100 to magnetic
10 tape 102.

The subject of the present invention is the error-correction decoding of data from a multi-track storage medium. Thus, the present invention is primarily
15 concerned with the operation of read circuitry 110 as it decodes data stored on magnetic tape 102 that is encoded with an error-correcting code.

Figure 2 is a block diagram depicting read circuitry in accordance with a preferred embodiment of the present invention. Note that the circuitry in **Figure 2** may also
20 be used for performing tasks associated with write circuitry 106 in **Figure 1** or other tasks within tape drive 100, as appropriate.

A processor 200 functions as the control center for read circuitry 110 (or tape drive 100, generally, in one
25 possible embodiment). Processor 200 communicates through memory 202 through bus system 204. Memory 202 holds a set of instructions that is executed by processor 200 to carry out the processes of the present invention described in subsequent Figures. Also connected to
30 device bus 202 is a read/write head interface 206, which

Docket No. 2001-010-TAP

allows processor 200 to control the reading and writing of data to/from magnetic tape 102 via read/write head 108. Data buffer 104, which is preferably some type of random-access memory (RAM) is also connected to bus system 204 to allow processor 200 to read and write data to/from data buffer 104. In an alternative embodiment, memory 202 and data buffer 104 are combined into a single bank of memory.

Processor 200 executes a set of instructions located in memory 202. As used in this application, the term "instructions" is to be interpreted broadly so as to encompass a wide variety of functional descriptive material that enables the processor to execute the processes of the present invention. In the conventional sense, instructions are a series of commands, that when executed in sequence by the processor, cause the processor to perform the processes of the present invention. A machine language program or a program written in a high-level computer language, such as C or C++ would include instructions, in this conventional sense.

For the purposes of this document, however, "instructions" are also taken to include a set of facts and rules of inference that, when processed by processor 200, cause processor 200 to follow a course of action that carries out the processes of the present invention. Such instructions could be written in a logic-programming or relational language such as Prolog. "Instructions" also include, for the purposes of this document, functions and expressions written in a functional

Docket No. 2001-010-TAP

programming language, such as Lisp. One of ordinary skill in the art will recognize that functional descriptive material of various other forms may be considered instructions for the purposes of the present document without departing from the scope and spirit of the present invention.

The depicted example in **Figure 2** and above-described hardware examples are not meant to imply architectural limitations. The actual hardware used to execute the processes of the present invention may vary.

Referring to **Figure 3**, a one-half inch wide length of magnetic tape 311 may contain up to 288 or more data tracks on multiple data stripes 312. A thin film magnetic read head is shown in upper position 313 and lower position 314 to read data from data tracks 312. If a tape read head has sixteen elements and, with movement of the head to multiple positions, each element can read nine tracks, then that magnetic read head could read 144 tracks. In order to read more tracks, such as 288 in the desired configuration, two data bands 315 and 316 are utilized. The tape head is movable to nine tracking positions in each of upper position 313 and lower position 314. That is, with the tape head in position 313 it can read 144 tracks in data band 315 and in position 314 it can read 144 tracks in data band 316.

In a preferred embodiment of the present invention, blocks of data are stored across tracks in encoded form using an error-correcting code, such as a Reed-Solomon code. This provides a level of data redundancy (in an information-theoretical sense) between tracks, such that

Docket No. 2001-010-TAP

if one or more of the tracks contains errors, redundant information in other tracks can be used to correct the error.

Figure 4 is a graphic representation of how an error correcting code, such as a Reed-Solomon code, can correct errors in a multi-track data storage medium, in accordance with a preferred embodiment of the present invention. Block **400** is a block of data. Block **400** comprises four units (e.g., words, bytes, nybbles, etc.) of raw data. Using an error-correcting code, these four units of data are transformed into block **402**, which contains six units of encoded data.

The key to an error-correcting code is that to recover the original data, not all of the encoded data is needed. Thus, if some of the encoded data gets corrupted or lost, the original data may still be recoverable if enough units of the encoded data are intact. In a multi-track data storage medium, it makes sense to place each of the units of encoded data on separate physical tracks, since physical corruption of the media may be confined to particular physical locations on the tape. Thus, some tracks of a multi-track medium may become corrupted while others remain unaffected.

For example, in **Figure 4**, any four out of the six units of encoded data are all that are needed to recover the original data. Thus, if units **401** and **403** are corrupted, units **404**, **406**, **408**, and **410** can be decoded to form decoded block **412**, which is identical to the original block **400**.

A careful examination of **Figure 4** reveals that there are two key pieces of information that are needed to reconstruct an original block of data from its encoded version. The first and most clearly needed piece of information that is needed is the encoded data itself. The second, not so clear, is the locations of the actual errors. For example, in **Figure 4**, we need to know that units **401** and **403** are corrupted before we can choose to decode the data using units **404**, **406**, **408**, and **410**.

It turns out that with many codes, the locations of the errors can be determined algorithmically, provided enough "good" (uncorrupted) data exists. In the case of a Reed-Solomon code, for instance, an extra unit of "good" data is needed in order to identify the location of an error.

Since magnetic tape and many other multi-track media store data and are read in linear sequential format, however, it is not possible to know with complete certainty where the errors in a given block of data are located. The present invention uses historical error correction data to make an "educated guess" as to which tracks are corrupted at any given location on the storage medium. Specifically, a preferred embodiment of the present invention selects a limited number of tracks with high past occurrence of errors to be treated as "erasures." Essentially, this means that these selected tracks are assumed to be corrupted and are ignored when the data is decoded.

What this means is that if one of these selected erasure tracks has an error occur on it at a particular

Docket No. 2001-010-TAP

point, less "good" data will be needed for decoding, since there is no need to identify where the error on the erasure track has occurred. The trade-off for this increase in efficiency is that if an erasure track is actually uncorrupted at a particular point, the uncorrupted data (which otherwise would be used for decoding) will be wasted. Thus, a preferred embodiment of the present invention limits the number of designated erasure tracks to those considered most likely to contain an error.

Figure 5 is a diagram illustrating how the processes of a preferred embodiment of the present invention may be applied to a multi-track data storage medium to maximize error-correcting capabilities. Chart **500** represents a length of multi-track magnetic tape. A plurality of tracks **502** span the width of the tape. At periodic locations along the length of the tape, blocks **504** of Reed-Solomon-encoded data are stored. Block **506**, for instance, is depicted as storing "n" units of Reed-Solomon encoded data (i.e., "n" Reed-Solomon coefficients).

Errors are present on the recording surface and are represented by black bars on the diagram. Error **510** is a small isolated error, whereas errors **512** are long portions of corrupted or erased data. At the end of the portion of tape, a checksum block **508** stores checksums, preferably some kind of cyclic redundancy check (CRC) value. Checksum block **508** can be used to identify data errors on tracks over a length of tape. Preferably,

Docket No. 2001-010-TAP

checksum blocks are present at periodic intervals along the length of the tape.

Checksum block 508 can detect that an error has occurred on a track over the length of tape, but it does not help in determining how extensive the errors are. In other words, checksum block 508 cannot be used to tell the difference between a small error such as error 510 and a large error such as errors 512.

What a preferred embodiment of the present invention does, however, to alleviate the situation is to calculate weights 516 for each track. These weights are preferably a measure of the number of errors corrected on each track as the track is read. In other words, in a preferred embodiment, each time an error on a given track is corrected, the weight for that track is increased. In this way the top "x" number of tracks in terms of highest weight can be designated as erasure tracks and thus ignored in the decoding process. Thus, long stretches of errors such as errors 512 will likely be treated as erasures, allowing more "good" data to be used for correcting isolated errors such as error 510.

As the storage medium is read from and the weights updated, the designated erasure tracks will change, depending on which tracks the most number of errors are occurring on at each particular position along the length of the storage medium. This allows a storage system to adapt to changing conditions as different physical locations along the medium are read. In a preferred embodiment of the present invention, these updates in

Docket No. 2001-010-TAP

erasure tracks are performed periodically, each time a checksum block is encountered.

In one embodiment of the present invention, all tracks that have contained errors at some point are treated as erasures until the number of errors and erasures at some point on the medium is so large as to make the errors at that point uncorrectable. In such case, low-weight erasures are removed from the set of designated erasures until all of the errors can be corrected.

Figure 6 depicts a process of decoding one block of Reed-Solomon encoded data from a multi-track magnetic tape in accordance with a preferred embodiment of the present invention. Block 600 includes eight Reed-Solomon coefficients (denoted "n" in the diagram, thus $n=8$). Block 600 represents two units of data (i.e., "m", the size of the original amount of data, is 2). Two tracks, and thus two of the coefficients, coefficients 602 and 608, are designated as erasures (i.e., "e" the number of erasures, is 2). Coefficient 602, which is treated as an erasure, is corrupted. Coefficients 604 and 606, which are not on erasure tracks, are also corrupted (i.e., "t" the number of errors [other than erasures], is 2). It is well known in the art that in the case of a Reed-Solomon code, if the following inequality holds, then the original data can be recovered:

$$2t + e \leq n - m.$$

In this case, $2t + e = 6 = n - m$, so the inequality is satisfied. Thus, the errors may be corrected. In addition, the weights for tracks 5 and 6 (corresponding

Docket No. 2001-010-TAP

to coefficients 604 and 606) will be increased, as errors were corrected on these tracks.

Figure 7 is a flowchart representation of a process of correcting errors in a multi-track data storage medium in accordance with a preferred embodiment of the present invention. A block of encoded data is read from the storage medium (in this case, a tape) (step 700). The block is decoded, while treating certain tracks designated as erasure tracks as containing corrupted data (step 702).

If any errors were corrected in decoding the block (step 704:Yes), the weight values for the tracks containing the errors are updated (step 706). If the end of a length of tape (containing a cyclic redundancy check or other suitable checksum) has not yet been reached (step 708:No), the process cycles to step 700 for further data reading.

If the end of the length has been reached (step 708:Yes), cyclic redundancy checks (or other suitable checksums) are calculated for the length of tape read (step 710). If there are no tracks that do not match (i.e., no errors occurred on the length of tape) (step 712:No), the process cycles to step 700 to continue reading data.

If, however, errors are detected (step 712:Yes), the "N" tracks with highest weight that contained errors in the previous length are identified (step 714). Those tracks are then designated as the new erasure tracks (step 716).

Figure 8 is a flowchart representation of a process of error correction code decoding a block of data for a multi-track medium using a variable number of erasures in accordance with a preferred embodiment of the present invention. The process described in **Figure 8** starts with a number of erasures being already defined. For example, all tracks that have previously contained errors may be designated erasure tracks at the beginning of the process described in **Figure 8**.

First, the number of designated erasures is checked to see if it exceeds the "order of the code" (step 800). In other words, the number of erasures is checked to see if it exceeds that amount which the code will allow to be decoded (the $n-m$ in the $2t-e \leq n-m$ equation described earlier in this document). If so (step 800:Yes), then the "x" highest-weight erasures are then designated current set of erasures and passed to the decoder (decoding software or hardware or a combination of both), with "x" being some number less than or equal to $n-m$ (step 802). If not (step 800:No), then the current set of erasures is simply passed to the decoder (step 804).

Next, the decoder attempts to decode the block of data from the multitrack medium (step 806). If the decoding was successful (step 808:Yes), then the process ends. If the decoding was not successful (step 808:No), a different "x" is tried or different coefficients are used for the decoding (step 810) and the process cycles to step 802, so as to (eventually) make an exhaustive search of all the decoding possibilities until the block is successfully decoded.

Docket No. 2001-010-TAP

Although the processes of the present invention have been described as being executed by an embedded stored-program computer system as in **Figure 2**, the processes of the present invention need not be carried out by such hardware. For example, **Figure 9** is a block diagram of "hard-wired" (special-purpose) hardware for carrying out the processes described herein. The apparatus described in **Figure 9** may be implemented in digital logic using conventional synthesis techniques.

Correction history registers **900** store weights for each track that has been corrected. These weights are fed into priority selector **902**, which selects the "X" highest weight erasures according to "X" input **904**, which is derived from decode logic **906**. Decode logic **906** decodes tape data **908** using the "X" highest-weight erasures as determined by priority selector **902**. Any adjustments to "X" necessary to ensure complete decoding of tape data **908** are made by transmitting a new "X" value to priority selector **902** through "X" input **904**. Properly decoded data is sent to output **910**, from which the data may be written to data buffer **104** (**Figure 1**).

It is important to note that while the present invention has been described in the context of a fully functional data processing system, those of ordinary skill in the art will appreciate that the processes of the present invention are capable of being distributed in the form of a computer readable medium of instructions and a variety of forms and that the present invention applies equally regardless of the particular type of signal bearing media actually used to carry out the

Docket No. 2001-010-TAP

distribution. Examples of computer readable media include recordable-type media such a floppy disc, a hard disk drive, a RAM, CD-ROMs, and transmission-type media such as digital and analog communications links.

5 The description of the present invention has been presented for purposes of illustration and description, and is not intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in
10 the art. The embodiment was chosen and described in order to best explain the principles of the invention, the practical application, and to enable others of ordinary skill in the art to understand the invention for various embodiments with various modifications as are
15 suited to the particular use contemplated.